分类号：_____密　级：_____
UDC：_____学　号：020923

东　南　大　学
硕　士　学　位　论　文

高维数据集的子空间聚类算法研究

研究生姓名：周晓云
导　师姓名：孙志挥　教授

申请学位级别　硕 士　学科专业名称　计算机应用
论文提交日期　2004.12　论文答辩日期　2004.12
学位授予单位　东南大学　学位授予日期　_____
答辩委员会主席_____评　阅　人　_____

二〇〇四年十二月

摘　要

作为数据挖掘中的一项重要技术，聚类分析具有广泛的应用领域。同时，聚类也是数据挖掘领域中一个相对比较困难的问题，而大规模、高维数据集的聚类算法已成为当前研究的热点。由于“高维诅咒”的存在，目前绝大多数算法在高维数据空间的情况下都无法得到理想的效果。此外，高维数据中含有的大量的随机噪声也会带来额外的效率问题。目前，子空间聚类算法是对大规模、高维数据集聚类的有效方法之一。

本论文的研究工作着重分析了传统聚类算法在处理大规模、高维数据集的困难和问题。比较了各种降维处理方法的优劣，从原理上论证了子空间聚类算法在处理大规模、高维数据集的优势。在此基础上，进一步对已有的子空间聚类算法—CLIQUE 进行了分析，指出 CLIQUE 算法存在的若干不足，即：CLIQUE 在划分网格时没有或者很少考虑数据的分布，在处理大规模、高维数据集时需进行大量的 I/O 访问操作，使效率大大降低。为此，我们在论文研究工作中提出了基于最优分割区间和数据划分的聚类算法（OpCluster）。同时考虑到高维数据集中有可能存在高维的聚类，而 CLIQUE 算法在低维聚类时将产生大量的候选项，还提出了基于单调递减阈值函数的有趣子空间发现算法（FIS）。

本文从理论和实验两方面证明了在处理大规模、高维数据集时上述两算法比 CLIQUE 更加高效。

关键词：数据挖掘、子空间聚类、阈值函数、高维、降维

Abstract

Clustering is an important data-mining technique used to find data segmentation and pattern information and is widely used in applications of financial data classification, spatial data processing, satellite photo analysis, and medical figure auto-detection etc. But clustering also is a difficult question in the field of data mining, especially dealing with large data set of high dimensionality. Most clustering algorithms, however, do not work effectively and efficiently in high dimensional space, which is due to the so-called "curse of dimensionality". In addition, the high-dimensional data often contains a significant amount of noise which causes additional effectiveness problems. Subspace clustering is an efficient method to do it at present.

In this paper, we analyze the problems and difficulties using traditional clustering algorithms dealing with the large data set of high dimensionality. We also compare the advantages and shortcomings with different dimensional reduction methods, and express the theory of subspace clustering. We find several shortcomings in the theoretical algorithm— CLIQUE through analyzing it. So we proposed an algorithm using the technique of optimal grids based on the data distribution and of data partition to deal with the large data set of high dimensionality more efficiently. We further research on the high dimensional cluster hidden in high dimensional data set, then give a new algorithm—FIS which find interest subspace based on threshold function which monotonically decreasing.

This paper proves that our two algorithms have better performance than CLIQUE in the work of mining the large data set of high dimensionality theoretically and experimentally.

Key Words: Data mining, Subspace clustering, Threshold function, High dimension, Dimensionality reduction

东南大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：周晓云　日 期：2006.12

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名：周晓云　导师签名：　日 期：

第一章 引言

二十世纪八十年代以来，人类产生和采集数据的能力迅猛增长。条形码技术在商品中的广泛应用、商业和政府事物的计算机化、加上数据采集工具的长足发展为人们提供了越来越多的数据。数以万计的数据库系统被应用到商业、政府、科学研究与工程技术等领域。此外，由于技术的成熟和价格的低廉，投入运行的数据库系统数量以越来越快的速度增加。数据和数据库应用的爆炸式增长促使人们产生对自动地从数据库中的大量数据寻找有用的信息和知识的迫切需求。在这种现实需求的驱动下，一个崭新的交叉研究领域出现了，这就是数据挖掘[1](Data Mining)，或者说数据库中的知识发现(KDD, Knowledge Discovery in Database)。

数据挖掘是从数据集中识别出有效的、新颖的、潜在有用的并最终可理解模式的非平凡过程[1]。其在功能上的强健性和应用领域的广泛性已被广为了解，十多年来，围绕这一主题所开展的深入而广泛的研究与探索工作常新不竭，成果丰硕。然而，Web 技术的兴起与人类基因组计划的实施使人们真正看到了什么是巨量数据—不论设备处理能力的增长如何显著，事实上存在的海量数据总使计算资源消耗殆尽！面对这样的海量数据，人们除了借助于高速计算机以外，就是从根本上研究并构造高效的知识抽取方法。

聚类[1]（Clustering）是当前数据挖掘领域得到广泛研究的问题之一。所谓聚类，就是将一个对象划分为一定数目的有意义的簇，使得同一个簇中的对象尽可能具有相同的特征，而属于不同簇的对象尽可能相异。聚类分析技术具有广泛的应用领域，如统计数据分析、模式匹配和图像处理以及商业领域等。大到地球和宇宙观察数据处理，小到生物基因结构分析，聚类算法都有其用武之地。

聚类是数据挖掘领域中一个相对比较困难的问题。大规模、高维数据的聚类尤其如此。尽管对聚类方法的研究已有数十年的历史，然而迄今为止，仍未有针对大规模、高维的数据进行聚类的有效算法。考虑到聚类技术拥有的广泛应用领域，在该领域的任何技术和方法上的创新与突破，都将不仅具有重要的理论价值，且具有显著的现实意义。

结合国家自然科学基金项目“基于数据挖掘技术的中观审计风险研究”（70371051）课题，我们进行了聚类算法的研究。在论文研究工作中，着重讨论了大规模、高维数据集中基于网格和密度的聚类算法—CLIQUE[3]，并根据此算法的局限性提出了改进算法。考虑到CLIQUE 在划分网格时没有或者很少考虑数据的分布，我们提出了基于数据分布的最优分割

面的网格划分方法，从而提高了算法的执行速度和结果精度。考虑到 CLIQUE 在大规模、高维数据集的情况下要进行大量 I/O 访问，使效率大大降低，提出了基于数据划分[1]的聚类发现算法，通过发现局部聚类来发现全局聚类。考虑到高维数据集中有可能存在高维的聚类，而 CLIQUE 算法在低维聚类时将产生大量的候选项，提出了基于单调递减阈值函数的有趣子空间发现算法（FIS）。

本文以下章节的内容安排如下：第二章介绍了聚类分析方法的基本概念和相关工作；第三章分析了高维数据聚类遇到的问题，给出了子空间聚类的原理和方法；第四章对于 CLIQUE 算法中几个不足之处，提出了改进算法—OpCluster，并给出了实验结果和性能评价；第五章提出了有趣子空间的定义，并在此基础上提出了基于单调递减阈值函数的有趣子空间发现算法--FIS，给出了实验结果和性能评价；第六章做出总结，指出本课题的一些不足和有待完善的方面以及下一步研究工作的方向。

第二章 聚类分析的基本概念

2.1 聚类的定义

聚类是数据挖掘中用来发现数据分布和模式的一项重要技术。聚类问题可以这样描述：将数据点集合分成若干类（称为簇，cluster），使得每个簇中的数据点之间最大程度地相似，而不同簇中的数据点最大程度地不同[1]。即：

定义 2.1 给定一个数据集合 $V\{v_1, v_2, \ldots, v_n\}$，其中 v_i（i=1,2,...,n）称为数据点，根据数据点间的相似程度将数据集合分成 k 组：$\{C_1, C_2, \ldots, C_k\}$，$C_i=\{v_{j1}^i, v_{j2}^i, \ldots v_{jt}^i\}$(i=1,2,...,k)，且 $\bigcup_{i=1}^{k} C_i = V$，的过程称为聚类。$C_i$（i=1,2,...,k）称为簇。

和一些数据挖掘技术不同，在进行聚类分析前用户一般并不知道数据集的特征。于是，从某种角度看，聚类分析是一种无指导的（unsupervised）学习过程。所以，在一个广义的数据挖掘过程中，聚类分析往往被作为最初的步骤，用于获得对于数据分布和聚集特性的初步了解，以在此基础上进行其他数据挖掘和规则发现的处理。

2.2 相似性测度

上述关于聚类定义的核心概念是数据点之间的相似性，即数据点之间在性态上的亲疏远近关系，它是数据聚类的根本依据。为了使聚类过程在定量的范畴上得以实现，必须恰当地对样本之间的相似性加以定义。聚类分析中，数据的相似性度量通常采用相似系数和距离加以定义。

(1) 相似系数

相似系数是描述数据对象之间相似性的测度，两个样本点之间愈相似，则其之间的相似系数愈接近 1；否则，两个样本点愈不相似，其之间的相似系数愈接近 0。根据给定的相似系数，可以构造一个反映数据相似形态的相似性矩阵。数据点之间的相似系数是满足如下条件的二元函数 $S: D\times D \to [-1, 1]$:

$$①\ |S(X,Y)| \le 1; \tag{2.1}$$

$$②\ S(X,Y) = S(Y,X); \tag{2.2}$$

$$③\ S(X,X) = 1. \tag{2.3}$$

条件③规定，一个数据点与其自身的相似性最大。在处理包含重复数据的数据集时，两个相互重合的数据点之间的距离也等于 1.

常用的相似系数包括数据点间的夹角余弦和相关系数：

① 夹角余弦：

$$C(X,Y)=\frac{\sum_i x_i \cdot y_i}{\sqrt{(\sum_i x_i^2)\cdot(\sum_i y_i^2)}}. \tag{2.4}$$

夹角余弦函数忽略两数据点(向量)间的绝对长度而考虑其在方向上的相互关系。两数据点间的相似性反映了其在状态空间方向上的一致性，当它们在方向上一致时，夹角余弦为 1；而在其之间毫无可比性(正交)时，其夹角余弦为 0。夹角余弦方法常用来描述概念型数据点之间的相似性关系。

② 相关系数：

$$R(X,Y)=\frac{\sum_i (x_i-\bar{x})\cdot(y_i-\bar{y})}{\sqrt{(\sum_i (x_i-\bar{x})^2)\cdot(\sum_i (y_i-\bar{y})^2)}} \tag{2.5}$$

相关系数是关于向量标准差的夹角余弦，它表示两个向量线性相关的程度。

(2) 距离

距离是刻划数据对象间差异性的测度，因此，数据对象之间距离的定义是有效地组织数据并加以聚类分析的前提条件。通常地，研究者并不孤立地对数据对象之间的距离加以定义，而是将所给定的具有 d 个相互独立的属性度的数据集映射到一个特定的 d 维线性空间 F^d 上，通过对空间 F^d 赋予某种距离的概念而在两个数据对象之间建立适当的距离关系。显然，数据对象之间的距离建立后，两者之间的距离越小，其相似性越大，否则相似性越小。

设数据集 $D\subset F^d$，F^d 上的二元关系 $dist: F^d\times F^d\to R$ 称为距离，如果 $dist(\cdot,\cdot)$ 满足：

① 非负性： $dist(x,y)\geq 0$， $dist(x,y)=0$ 当且仅当 $x=y$； (2.6)

② 对称性： $dist(y,x)=dist(x,y)$； (2.7)

③ 三角不等式： $dist(x,y)\leq dist(x,z)+dist(y,z)$，或 (2.8)

④ $dist(x,y)\leq \max\{dist(x,z),dist(y,z)\}$ (2.9)

其中，满足条件①, ②, ③的距离是通常意义下的距离，而满足①, ②, ④的距离函数称为极端距离，它在某些聚类模型构造中具有重要的应用。以下是聚类分析中常见的有关距离的定义：

① 明科夫斯基(Minkowski)距离（λ_p-距离）：

$$d_p(X,Y)=(\sum_i |X_i-Y_i|^p)^{1/p},\quad X,Y\in F^d,\quad p=1,2,... \tag{2.10}$$

上式中，当 p=1 时，为绝对值距离(又称 Manhattan 度量或网格变量)；当 p=2 时，称为欧氏距离；而当 $p=\infty$ 时，$d_\infty(X,Y)=\max_i |X_i-Y_i|$，称为切比雪夫(Chebyshev)距离。

② 马氏 (Mahalanobis) 距离：

$$d(X,Y)=\sqrt{(X_i-X_j)^T\Sigma^{-1}(X_i-X_j)},\quad X,Y\in F^d \tag{2.11}$$

其中 Σ 为数据集 D 所构成的数据矩阵的协方差矩阵，它是样本总体分布的协差估计量。

马氏距离的优点在于其在线性变换下的不变性，这一性质使马氏距离克服了明氏距离定义受量纲影响的缺陷。同时，马氏距离还可以有效地处理具有多重相关性的数据。但是，协差矩阵的构造与计算是十分复杂的，因此，这种方法在处理大规模数据时并不适用。

③ 兰氏(Canberra)距离

$$d(X,Y)=\sum_i \frac{|X_i-Yi|}{|X_i+Yi|},\quad X,Y\in F^d. \tag{2.12}$$

同样地，兰氏距离也可以克服数据集在量纲上不一致的缺陷，而且在计算复杂度上较之马氏距离简便。

④ 海明(Hamming)距离

和上述面向数值型数据的距离不同，海明距离在两个等长的字符串之间定义距离：

$$d(S_1,S_2)=count_i(S_{1i}\neq S_{2i}) \tag{2.13}$$

其中，S_1,S_2 为两个字符串，S_{1i},S_{2i}，i=1,2,... 为其在各位上的码字。海明距离为字符型数据聚类分析提供了基础条件，它常用于文本聚类问题。

此外，聚类分析还常采用斜交空间距离和卡方距离来对数据点之间的距离加以定义。由于选择不同的距离定义方法所获得的聚类结果会有所差异，因此，在实用中，可以采用几种距离进行计算、对比，选择一种较为合理的距离进行聚类。

2.3 聚类的定义方法

聚类分析是面向应用的技术，因此，聚类的定义与待处理的数据类型有关。基于不同的模型构造思想，学术界提出了一系列常见的定义[24]。

(1) 基于距离的定义(Distance-based)：一个聚类是这样一组数据的集合，其全体成员彼此之间的距离的最大值均小于其与非该聚类成员之间的距离的最小值。显然，这种定义方法过于机械，它对于刻划复杂数据聚类的能力较弱。

(2) 基于质心的定义(Centroid-based)：在数据集合中指定若干个质心，根据各数据点到这些质心的距离划分聚类，它使得一个聚类中的元素到其质心的距离小于其到任何其它质心的距离。这种定义方法的显著缺陷在于它不能够适应于具有不规则形状的聚类。

(3) 基于连接的定义：一个聚类是一组彼此连接的数据点的集合，其每个元素到聚类中其他元素之间的接近程度大于其到非聚类成员之间和接近程度。这种定义方法的有效性在于它能够描述任意形状的聚类，但当数据集包含噪声时，基于连接的定义方法难以区分夹杂于噪声数据之间的不同聚类。

(4) 基于密度的定义：聚类是数据空间中的稠密区域元素所构成的子集合，两个聚类之间的边界通过稀疏区域划定。基于密度的定义能够适应聚类形状不规则和包含噪声数据的情形，因此它较之其他定义方法具有独到的优越性。

(5) 基于相似性的聚类：聚类是一组“相似”数据对象的集合，处于不同聚类的元素彼此互不“相似”。这里，数据对象之间的相似度由相似系数确定，而对象之间是否相似则必须通过预设的阈值加以规定:相似系数大于阈值的对象之间是相似的，否则就不相似。

2.4 聚类分析的过程

聚类分析是一个基于数据的建模过程，其处理的对象是数据集，聚类的结果是关于数据聚类模式的知识。聚类分析过程一般包括数据准备阶段、特征生成阶段和聚类发现阶段。

2.4.1 数据准备

数据准备阶段的工作包括数据采集、数据清洗和数据标准化。其中数据清洗的任务包括剔除空值和噪声、修正错误数据等。该阶段的另一个任务是数据的标准化，由于对象数据集通常具有多个不同的属性要素，它们在取值上往往采用不同的单位或量纲，因而导致取值幅度水平上的差异并最终影响聚类的结果。因此，为了使数据样本在各个属性上的取值在量纲上一致，通常在进行聚类分析之前需要对数据进行归一化(或标准化)处理。常见的数据归一化方法有[24]：

(1) 最大—最小归一化法：将数据集在每一属性分量的取值除以该属性上取值的绝对值最大值：

$$x_{ij}^{'} = x_{ij} / \max_{i} |x_{ij}| . \tag{2.14}$$

归一化后的数据取值介于-1到1之间。该方法对于具有类均匀分布的数据归一化效果较好，而当数据集包含离群的噪声数据时则不够理想。

(2) 总和标准化：将数据点的各分量除以全体数据在该分量上的取值之和：

$$x'_{ij} = x_{ij} / \sum_{i=1}^{N} x_{ij}, \quad i = 1,2,...,N, \quad j = 1,2,...,d. \tag{2.15}$$

这种标准化方法所得的新数据集满足：$\sum_{i=1}^{N} x'_{ij} = 1, \quad j = 1,2,...,d$.

(3) 均值标准差标准化，即：

$$x'_{ij} = (x_{ij} - \mu_j) / \sigma_j, i = 1,2,...N, \quad j = 1,2,..d. \tag{2.16}$$

其中μ_j，σ_j分别为第j列全部数值的均值和标准差。采用这种方法所得标准化后的数据满足：

$$\mu'_j = \frac{1}{N}\sum_{i=1}^{N} x'_{ij} = 0, \quad \sigma'_j = (\frac{1}{N}\sum_{i=1}^{N}(x'_{ij} - \mu'_j)^2)^{1/2} = 1. \tag{2.17}$$

均值标准差标准化方法特别适于对符合正态分布的数据，处理后的绝大多数数据值将位于-1到1之间。

(4) 极差标准化，即：

$$x'_{ij} = (x_{ij} - \min_i\{x_{ij}\}) / (\max_i\{x_{ij}\} - \min_i\{x_{ij}\}), \quad j = 1,2,...,d \tag{2.18}$$

经过这种标准化所得的新数据集在各分量上的极大值为1，极小值为0，其余的数值均在0与1之间。

2.4.2 特征生成

特征生成阶段的任务是计算并提取最终用于聚类的数据特征，恰当地择取能够揭示所给数据集有趣知识的属性字段。在这一阶段，研究者根据自己的经验和对数据的认识选择描述数据间相似性的方法，构造数据间的相似关系，如生成相似矩阵、构造密度函数等。数据集的特征提取有时是基于数据本身的，如基于连接的聚类定义方法通过直接运用数据点的物理位置关系建立聚类。在更多的情况下，这一过程是在将原始数据集向特定的空间映射后完成的，如基于相似性的方法，基于密度的方法等。基于相似性的方法在原始数据间建立相似性测度，使本来可能是属于定性的关系转化为定量的关系。基于密度的方法将原始数据集映射为其所在空间上的密度函数，形象地说，数据样本在原始空间中的疏密分布特征被转化为分布函数在相应空间上的凸凹不平的曲面。特征生成阶段为运用聚类算法发现聚类提供了必要

的解析手段。

特征生成阶段的另一项工作是数据的降维处理。在对高维数据集进行聚类分析时，为了避免高维诅咒(Curse of dimensionality)的出现，有时需要对数据进行降维处理。降维处理的方法之一是依据下述准则删除数据集中具有以下特征的属性：

(1) 数据在某一属性上的取值对于反映数据聚类模式没有作用或作用较小；

(2) 一个属性与其它属性或属性组之间具有明显的相关性。

降维处理可以使聚类分析能够聚焦于那些能够有效揭示数据聚类特征的属性集。另一种降维处理的方法是投影降维方法。这种方法通过投影运算将高维空间中的数据集映射成具有较低维数的数据集。其降维的依据通常采用数据矩阵的特征值分解(eigenvalue decomposition)或奇异值分解(singular value decomposition)方法

2.4.3 聚类模式发现

在上述两个阶段工作的基础上，本阶段具体完成聚类的发现、验证、评价和优化工作。为了真实地反映隐藏于数据间的聚类模式，减少聚类模型与实际系统之间的误差，选择合理的聚类算法至关重要。目前，没有一种聚类方法可以适应所有类型的数据，对于同一个数据集，不同的聚类方法得到的聚类结果也不尽相同。实际上，聚类分析是一个通过用户的介入实现自动性和交互性的多阶段过程，其中每个步骤都至关重要。在下节的讨论中，我们将就各种现有的聚类算法及其性质加以论述。

聚类模式发现的过程通常是人机交互的过程，用户决定具体的算法选择，对算法所涉及的有关参数和阈值加以赋值，对聚类的结果加以评价和解释。通过不断的尝试和修正，用户将最终获得关于数据聚类的知识。随着可视化技术的日益成熟，现代聚类分析过程通常是在良好的可视化环境下完成的，显然，它对于快速准确地解决聚类分析问题具有巨大的帮助。

目前，聚类分析应用正从面向小规模、低维数据集向面向海量、高维数据集转移。随着数据维数的提高，一些本来对处理低维数据集有效的算法不再有效，而高效的能够恰当处理高维数据的算法尚不多见。此外，随着数据规模的膨胀，一些能够“精致地”处理小规模数据的聚类算法也不再适应，人们迫切需要能够有效处理上述两个问题的高效聚类算法出现。

2.5 聚类的主要方法

本节将首先讨论几种主要的聚类方法，包括分割聚类方法、层次聚类方法、基于密度

的聚类方法和基于网格的聚类方法，然后对其他聚类方法作概括性的介绍。

2.5.1 分割聚类方法

分割聚类方法[1]（Partitioning Clustering Method） 是发展比较早的一大类聚类分析方法。它是一种基于原型（Prototype）的聚类方法，其基本思路是：首先从数据集中随机地选择几个对象作为聚类的原型，然后将其他对象分别分配到由原型所代表的最相似、也是距离最近的类中。对于分割聚类方法，一般需要一种迭代控制策略，对原型不断地进行调整，从而使得整个聚类得到优化，例如使得格对象到其原型的平均距离最小。

根据所采用的原型的不同，代表性的分割聚类方法主要包括 k-means 和 k-medoid 两类。

（1）k-means 算法

假设有 n 个对象需要分成 k 类，那么在 k-means 算法中，首先随机地选择 k 个对象代表 k 个类，每一个对象作为一个类的中心，根据距离中心最近的原则将其它对象分配到各个类中。在完成首次对象的分配之后，以每一个类中所有对象的各属性均值（means）作为该类的新的中心，进行对象的再分配，重复该过程直到没有变化为止，从而得到最终的 k 个类。

k-means 算法中，聚类的个数 k，是必须预先指定的参数。聚类的过程可以通过下述几个步骤来描述：

1. 随机地选择 k 个对象，每一个对象作为一个类的“中心”，分别代表将分成的 k 个类。
2. 根据距离“中心”最近的原则，将其他对象分配到各个相应类中。
3. 针对每个类，计算其所有对象的平均属性值，作为该类新的“中心”。
4. 根据距离“中心”最近的原则，重新进行所有对象到各个相应类的分配。
5. 若 4 中得到的新的类的划分与原来的类的划分相同，则停止计算。否则，转 3。

另外，k-means 算法还有一种扩展算法，称为 k-modes 算法，该算法是将 k-means 算法的思想应用于分类变量的情况。

（2）k-medoid 算法

在 k-medoid 算法中，首先选择 medoids 作为各个类的原型，在根据距离 medoid 最近的原则将其他对象分配到各个类中。那么，什么是 medoids 呢？假设有 n 个对象需要分成 k 类，那么 medoids 是分别接近于 k 个类的中心，并且按照一定的标准使聚类的质量达到最好的 k 个对象。比较著名的 k-medoid 算法有 PAM（Partitioning Around Medoids）算法、CLARA（Clustering LARge Applications）算法[25]及 CLARANS（Clustering Large Applications based on

RANdomized Search）算法[26]。

PAM 算法是比较基本的 k-medoid 类型的算法。在 PAM 算法中，最为关键的是寻找这 k 个代表对象。对此，PAM 算法首先随机选取 k 个对象作为 k 个类的代表 medoid，并将其它对象分配到与其距离最近的 medoid 所代表的类中。然后按照一定的质量检验标准选择一个 medoid 对象和另一个非 medoid 对象进行交换，使得聚类的质量得到最大限度的提高。重复上述对象交换过程，直到质量无法提高为止，并将此时的 k 个 medoids 作为最终的 k 个 medoids，进行非 medoid 对象的分配，形成最终的聚类。

CLARA 算法是比较早的立足于处理大数据集的一种算法。在 CLARA 算法中，类的代表对象 medoids 不是从整个数据集中选择的。该算法首先从整个数据集中抽取一个样本，然后针对样本集应用 PAM 算法寻找类的代表对象 medoids。如果样本的抽取比例较适合，那么，既可以减少计算量，又基本不影响聚类的质量。

CLARANS 算法也是对 PAM 算法的一种改进，但是改进的方法与 CLARA 算法有所不同。同 CLARA 算法类似，CLARANS 算法也采用抽样的方法减少数据量，并采用 PAM 算法寻找代表对象 medoids，但是抽样的内容和寻找 medoids 过程与 CLARA 算法不同。CLARA 算法在固定的样本中寻找代表对象 medoids，进行 medoid 对象和非 medoid 的替换；而 CLARANS 算法寻找代表对象 medoids 并不局限于样本集，而是在整个数据集中随机抽样寻找。CLARANS 算法是从数据挖掘的角度提出的比较早的聚类算法之一。实验结果表明 CLARANS 算法比 PAM 算法和 CLARA 算法更有效。

总体来说，在聚类的形状为凸形，大小和密度相似，并且聚类的数目可以合理估计的情况下，上述各种分割聚类算法都是比较有效的，能够形成合理地聚类结果。

2.5.2 层次聚类方法

层次聚类方法（Hierarchical Clustering Method）也是发展比较早应用比较广泛的一大类聚类分析方法，它是采用“自顶向下（Top-Down）”或“自底向上（Bottom-Up）”的方法在不同的层次上对对象进行分组，形成一种树形的聚类结构。如果采用“自顶向下”的方法，则称为分解型层次聚类法（Divisive Hierarchical Clustering）；如果采用“自底向上”的方法，则称为聚结型层次聚类法（Agglomerative Hierarchical Clustering）。

层次聚类方法同分割聚类方法的不同之处在于：对于分割聚类方法而言，一般需要一种迭代控制策略，使得整个聚类逐步优化；层次聚类方法并不是试图寻找最佳的聚类结果，

而是按照一定的相似性判断标准，合并最相似的部分，或是分割最不相似的两个部分。如果合并最相似的部分，那么从每一个对象作为一个类开始，逐层向上进行聚结；如果分割最不相似的两个部分，那么从所有的对象归属在唯的一个类中开始，逐层向下分解。

在层次聚类方法中，判断各个类之间的相似程度的准则是：假设 C_i 和 C_j 是聚结过程中同一层次的两个类，n_i 和 n_j 分别是 C_i 和 C_j 两个类中的对象数目，$p^{(i)}$ 为 C_i 中的任意一个对象，$p^{(j)}$ 为 C_j 中的任意一个对象，f_i 为 C_i 中对象的平均值，f_j 为 C_j 中对象的平均值，那么下列四种距离表示比较广泛地应用于计算两个类之间的差异度：

（1） 平均值距离：$d_{\max}(C_i,C_j)=d(f_i,f_j)$ (2.19)

（2） 平均距离：$d_{average}(C_i,C_j)=\frac{1}{n_i n_j}\sum_{p^{(i)}\in C_i,p^{(j)}\in C_j} d(p^{(i)},p^{(j)})$ (2.20)

（3） 最大距离：$d_{\max}(C_i,C_j)=\max_{p^{(i)}\in C_i,p^{(j)}\in C_j} d(p^{(i)},p^{(j)})$ (2.21)

（4） 最小距离：$d_{\min}(C_i,C_j)=\min_{p^{(i)}\in C_i,p^{(j)}\in C_j} d(p^{(i)},p^{(j)})$ (2.22)

比较传统的层次聚类基本算法有 AGNES(AGglomerative NESting)算法和 DLANA（DIvisive ANAlysis）算法，它们分别为聚结型层次聚类算法和分解型层次聚类算法。传统的层次聚类算法比较简洁，易于理解和应用，但是聚类的结果受各个类的大小和其中对象分布形状的影响，适用于类的大小相似且对象分布为球形的聚类。在对象分布形状比较特殊的情况下，可能会产生错误地聚类结果。另外，每一次类的聚结或分解都是不可逆的，并直接影响着下一步地聚结或分解。如果某一步的聚结或分解不理想，形成聚类的质量就可能很低。

近几年，出现了一些新的属于层次聚类方法地聚类算法，一般采用的是聚结型层次聚类策略，例如：BIRCH（Balanced Iterative Reducing and Clustering using Hierarchies）算法[2]、CURE(Clustering Using Representatives)算法[22]、ROCK[27]算法和 Chameleon 算法[7]等。

BIRCH 算法是一种综合优化的多阶段聚类技术，它的核心是采用了一个三元组的聚类特征树（CF 树）汇总了一个簇的有关信息，从而使一个簇的表示可以用对应的聚类特征，而不必用具体的一组点表示，通过构造分支因子 B 和簇直径阈值 T 来进行增量和动态聚类。该方法通过一次扫描就可以进行较好聚类，比较适合于大型数据集。但它只适合于类的分布呈凸状或球状情况，需要提供正确的聚类数和对簇直径 T 的仔细选择，不适于高维数据。

CURE 算法采用了基于质心和基于代表对象方法之间的中间策略。它不用单个质心或对象来代表一个簇，而选择数据空间中固定数目的具有代表性的点，并将这些点乘以一个适当

的收缩因子，使它们更靠近簇的中心。选择多个代表使得该算法可以适应非球状的几何形状，簇的收缩或凝聚可以有助于控制噪声的影响。同时该方法采用了随机抽样与分割相结合来提高效率，对大型数据库有良好的伸缩性。CURE 算法首先将输入的每个点作为一个聚类，然后合并相似的聚类，直到聚类的个数为 k 时为止。在 CURE 中指出，基于中心点的方法和所有的点的距离计算方法都不适合非球形或任意形状的聚类，因此 CURE 采用了折衷的方法，即用固定数目的点表示一个聚类，从而提高了算法挖掘任意形状的聚类的能力。CURE 算法的时间复杂性为 $O(n.n)$(低维数据)和 $O(n.n.\log n)$(高维数据)，算法在处理大量数据时必须基于抽样、划分等技术。

ROCK 是用于对分类变量属性进行聚类的聚结型层次聚类算法。该算法首先构筑一个稀疏图（Sparse Graph），然后采用互连度（Interconnectivity）度量两个类之间的相似度。而互连度的计算依赖于不同的类拥有共同邻居（Neighbor）的数据点的数目。

Chameleon 算法是在 ROCK 算法的基础上提出的，它采用互连度（Interconnectivity）和接近度（Closeness）来度量两个类之间的相似性。在聚类的过程中，如果两个类之间的互连度和接近度与类的内部对象间的互连度和接近度高度关联，则进行合并。该算法处理不规则形状聚类的能力比较强。

2.5.3 基于密度的聚类方法

基于密度的聚类方法(Density-Based Clustering Method) 以局部数据特征作为聚类的判断标准。类被看作是一个数据区域，在该区域内对象是密集的，对象稀疏的区域将各个类分隔开来。多数基于密度的聚类算法形成的聚类形状可以是任意的，并且一个类中对象的分布也可以是任意的。

基于密度的聚类方法研究是非常活跃的一个领域，近几年提出了许多新的算法，例如：1996 提出的 DBSCAN(Density-Based Spatial Clustering of Applications with Noise)算法[15]、1998 提出的 WaveCluster 算法[6]、DENCLUE(DENsity-based CLUstEring)算法[4]、CLIQUE(Clustering In QUEst)算法[3]和 1999 提出的 OPTICS(Ordering Points To Identify the Clustering Structure)算法[16]等。其中 WaveCluster 算法、CLIQUE 算法在基于密度的同时，也是基于网格的。

DBSCAN 算法是在数据集上定义一种密度可达等价关系，对应的划分就是聚类。密度可达关系定义为:如果一个对象的 ε-领域至少包含了最小数目的 Minpts 个对象，该对象为核

心对象；如果 p 是在核心对象 q 的 ε-领域,则称对象 p 为对象 q 的直接密度可达；如果存在对象链 $p_1, p_2, \ldots, p_n$，$p_1=q, p_n=p$，对 $p,p_i \in D$ (对象集)，p_i+1 是从 p_i 关于 ε 和 Minpts 的直接密度可达，则对象 p,q 是关于 ε 和 Minpts 密度可达的。此方法是通过不断执行区域查询来实现聚类，对 ε 和 Minpts 相对敏感，且两参数难以确定。DBSCAN 基本思想是：对于一个聚类中的每一个对象，在其给定半径的邻域中包含的对象不能少于某一给定的最小数目，然后对具有密度连接特性的对象进行聚类。在该算法中，发现一个聚类的过程是基于这样的事实：一个聚类能够被其中的任意一个核心对象所确定。DBSCAN 算法可以挖掘任意形状的聚类，对数据输入顺序不敏感，并且具有处理异常数据（噪音）的能力。该算法的时间复杂性为 $O(n^2)$；在空间索引如 R*-树的支持下,其复杂性为 $O(n\log n)$。需要指出的是，DBSCAN 算法没有考虑建立索引的时间，而建立索引通常需要消耗大量的时间。

OPTICS 算法是在 DBSCAN 算法的基础上提出来的，该算法并不显式地生成数据类，而是基于密度建立对象的一种排序，通过该排列给出对象的内在聚类结构，通过图形直观的显示对象的分布即内在联系。OPTICS 算法的基本结构同 DBSCAN 算法的基本结构是一致的。

DENCLUE 算法利用密度分布函数通过识别密度吸引子（Density Attractor）的方法进行聚类。密度吸引子是密度函数的局部极值点。

2.5.4 基于网格的聚类方法

基于网格的聚类方法采用一个多分辨率的网格数据结构。它将空间量化为有限数目的单元，这些单元形成了网格结构，所有的聚类操作都是在网格上进行。这种方法的主要优点是处理速度快，其处理时间独立于数据对象的数目，仅依赖于量化空间中每一维上的单元数目。

基于网格方法的有代表性的算法包括：STING(STatistical INformation Grid)算法[5]、WaveCluster 算法[6]和 CLIQUE(Clustering In QUEst)算法[3]。

STING 算法基于多分辨率聚类技术，它将空间区域划分为矩形单元。针对不同级别的分辨率，通常存在多个级别的矩形单元，这些单元形成了一个层次结构：高层的每个单元被划分为多个低一层的单元。关于每个网络单元属性的统计信息（例如平均值、最大值和最小值）被预先计算和储存。

WaveCluster 算法是一种基于多分辨率变换的聚类方法,它首先在数据空间上强加一个

多维网格结构来汇总数据，然后采用一种小波变换来变换原特征空间，在变换后的空间找到聚类区域。由于小波变换的特性使该算法具有很多优点：计算复杂度为 O(n)，发现任意形状的簇，成功处理孤立点，对输入顺序不敏感，领域独立，可以处理多达 20 维的数据。

CLIQUE 算法是基于网格和密度聚类算法的又一个典型代表，由于该算法的主要思想和方法是本文研究和参考的重点，在以后章节中还需对该算法加以介绍，在此从略。

除了上面提到的分割聚类方法、层次聚类方法、基于密度地聚类方法和基于网格的聚类方法以外，还可以将相关的数学模型用于聚类，例如统计学方法和神经网络方法等。

必须指出在聚类算法的研究和应用中，就某一个聚类方法而言，往往是多种聚类思想融合的结果，并不能简单地将其归为上述某一类算法。而且，不同的聚类方法也可以进行结合，结合后的算法可能更有效。

2.5.5 聚类算法性能评价

随着聚类应用的普及和研究工作的深化，基于各种思想的聚类算法不断出现。但是，没有一种算法可以适应于所有类型的数据。所以，对于具体的应用目的，了解并选择适当的聚类算法十分关键。在考虑具体的聚类算法时，通常依据以下的评价准则来衡量一种聚类算法的优劣性及适应性[28]：

(1) 针对大数据集的处理能力

数据挖掘所面对的一般都是大数据集，需要从中挖掘出聚类知识信息。而多数聚类分析算法只适用于处理数据量比较小的情况，例如几十个或者上百个数据对象，而不适用于处理数据量更大的情况。通过抽取样本降低数据量是比较常见的处理大数据集的方法。但是，在样本选择不当的情况下，通过样本对象形成地聚类结果同实际的聚类可能存在着较大的偏差。因此，对大数据集的处理能力仍然是聚类算法研究的一个重要内容。

（2）高属性维数据处理能力

具有对高属性维数据的处理能力也是聚类算法研究的一个重要内容。高属性维数据是比较常见的一种数据形式，而许多聚类算法在属性维数比较低的情况下能够生成质量比较高地聚类结果，却难以应用于高属性维数据的情况。这方面的研究也是当前的一大热点。

（3）对象分布形状不规则的处理能力

多数聚类算法适用于处理各个类的对象分布为球形，且各个类的大小基本相同的情况。在类的大小差别比较大，或对象的分布为非球形，例如椭圆形、连接成串型及其他更复杂的

形状或混合形状，许多算法会形成错误的聚类结果。而实际聚类问题中对象的分布形状可能是任意的，因此提高对不规则形状的处理能力也是聚类知识发现的重要研究内容。

（4） 异常值的处理能力

异常值，有时也称为噪声，指的是对象取值远远偏离于常见取值的情况。出现异常值是在数据集中的普遍存在的现象，这可能是由于主观错误引起的，也可能是客观实际而产生，一般情况下难以避免。异常值的存在往往会给聚类的结果造成比较大的影响，甚至产生错误的聚类结果。

（5） 对数据输入顺序的独立性

有一些算法的聚类结果会受到数据输入顺序的影响，也就是说，同一个数据集应用同一个聚类算法在数据输入顺序不同的情况下会产生不同的聚类结果。这很显然是算法不理想而存在缺陷的一方面。

（6） 减少对先验知识或参数的依赖性

许多聚类算法在运行之前，需要输入一定的参数，例如：需要生成的聚类的个数、样本的大小、代表对象的数目、基于密度的聚类算法中有关密度的参数等。这些参数的设置往往对聚类的结果有着非常大的影响。参数设置得合适，会得到比较满意的聚类结果；参数设置的不合适，则会产生不同的、甚至是错误的聚类结果。由于在得到聚类结果之前，可能形成的聚类的数目、大小、对象的分布情况等相关信息都是未知的，要求用户合理地给出聚类算法所要求的参数，往往比较困难的。

（7） 聚类结果的表达与理解

在进行数据挖掘之前，对象类的划分是未知的。在数据挖掘之后，需要将聚类的结果以用户可以理解的方式表达出来，并进行合理的分析与解释。在属性维数比较小的情况下，比较容易采用可视化的方法表达聚类的结果。但是，在属性维数比较高的情况下，聚类结果的表达和理解就比较困难。

第三章 子空间聚类

3.1 高维空间聚类问题

近年来，聚类分析技术在空间数据库、基因分析、WEB 知识发现等领域的应用使人们进一步认识到高维空间数据聚类问题的重要性和特殊性。在这些应用场合，描述数据对象特征的属性集很大，转化后生成的数据集在维数上可达到数百个之多，这种超乎常规的高维度数据给现有聚类算法的成功应用造成了难以逾越的困难[24]。

当数据维数很高时，数据的分布规律经常呈现出反常的特征，如噪声水平提高、数据密度稀疏、常规意义下的距离定义不再有效等，这种高维异常现象在数据挖掘中被称为“高维诅咒（Curse of dimensionality）”。由于这种异常现象的出现，多数在较低维数情况下性能优越有效的聚类算法不再有效，即高维失效效应。这是因为，在低维数据空间中最有效的聚类算法一般都是基于某种空间层次数据结构、数据分割或空间网格结构的，它们在空间维数较低时，才具有较高的存取效率。例如，基于数据分区的索引结构在高维空间中其性能急剧下降，使得采用这种结构的聚类算法（如 DBSCAN）效率明显下降甚至不能运行。对于基于网格的聚类算法，由于网格单元的数量随维数的增加呈指数级增长，使算法的空间和时间复杂度显著增加。一般来说，对于索引结构，当数据维数超过一定的值时，数据存取操作和关键指标的计算将需要访问所有索引结构，此时算法的性能还不如顺序扫描的处理方法有效。

3.1.1 高维数据中噪声的影响

在大规模数据集的聚类中，噪声的影响是最主要的问题之一，在高维情况下，这个问题将变得更加严重。在大量噪声的高维数据集情况下，任何聚类算法都不可能以线性的时间复杂度发现聚类[11]。

定理 3.1[11] 在含有噪声的高维数据集中正确发现聚类最坏情况的时间复杂度是非线性的。

证明： 不失一般性，假设数据集中含有 O(n)噪声。在最坏情况下，一开始就读入所有噪声数据，这实际上意味着在这期间不可能得到任何噪声。但事实上，处理系统并不知道所读的这些数据是噪声而不属于任意聚类。因此，在接下来读剩余的数据时，必然会将所读数据与先前的噪声数据比较，计算它们的相似性，每一次计算相似性的时间复杂度不可能是 O(1)。因含有 O(n)噪声，所以含有噪声的高维数据空间中，正确发现聚类的时间复杂度是非线性

的。

3.1.2 高维数据对距离测度的影响

聚类的核心概念是相似度，而许多传统的聚类算法，如 k-means，BIRCH 等都用到了距离来表示两个对象间的相似度。在高维数据空间的情况下，距离测度将是毫无意义的。由于数据量不可能随着维数的增加而指数级增加，因此在高维空间中，存在着大量的稀疏部分。其结果是，最远的两个数据点之间的距离和最近的两个数据点间的距离差别将很小，或者几乎没有[11]。

3.1.3 高维数据对网格的影响

相比低维数据，高维数据空间很难确定一个聚类的中心。在基于网格的聚类方法中，聚类往往被一些(d-1)维的分隔平面所分割。不妨简单假设在每一维上只分割一次，这样就有 d 个(d-1)超平面将整个数据空间分成 2^d 个单元格，原有的数据之间的相邻属性将消失。下面提供一种最坏情况。设所有数据在$[0,1]^d$ 中，每一维在 0.5 的地方被分割。数据分布在以(0.5,0.5,...,0.5)为球心，半径为 ε ($0<\varepsilon 1$)的超球体中。当 d>20 时，某个聚类中的绝大多数数据点将被分割到各个单元格中。因此为了发现这个聚类，我们可能要搜索 2^d 个相邻单元格，计算量将非常巨大[11]。

3.2 降维处理

高维数据的降维技术包括特征转换和特征选择。特征转换通过综合原来属性维把数据集在较少的属性维上描述，如主成分分析（principle component analysis）、奇异值分解（singular value decomposition）。这些技术已经比较成功的应用于数据集潜在结构的发现。但是，由于它们保存了数据对象之间的相对距离特征，使得处理系统的执行效率很低，尤其是当聚类隐藏在许多不相关的属性维和噪声数据中。同时，对于综合原有属性特征而生成的新的属性特征，对用户来说是相当难以理解的。特征选择方法是通过选择相关性最高的属性特征，剔除那些相关性不高的属性特征，并在这些属性特征组成的空间中来发现聚类。尽管特征选择同样在某些数据集上得到了成功的应用，但是当为了在不同的子空间发现聚类时，特征选择方法就显得非常困难了[20]。

3.3 子空间聚类的原理

子空间聚类是特征选择的扩展，旨在发现同一数据空间不同子空间中的聚类。与特征选择类似，子空间聚类需要一个搜索算法和一个评价指标。此外，子空间聚类必须知道如何限制评价指标所涉及的范围，以便考虑不同聚类的不同子空间。

3.3.1 目的

我们将用一个简单的数据集来说明子空间聚类的必要性。参照文献[20]，我们产生一个数据集，在 3 个属性维中包含 400 个数据。数据集包含 4 个聚类，每个聚类只存在于 2 个属性维中，各 100 个数据。如图 3.1 所示

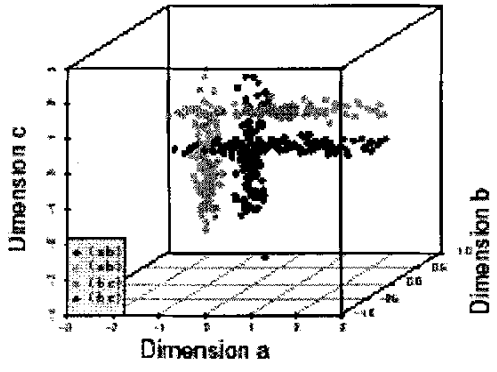
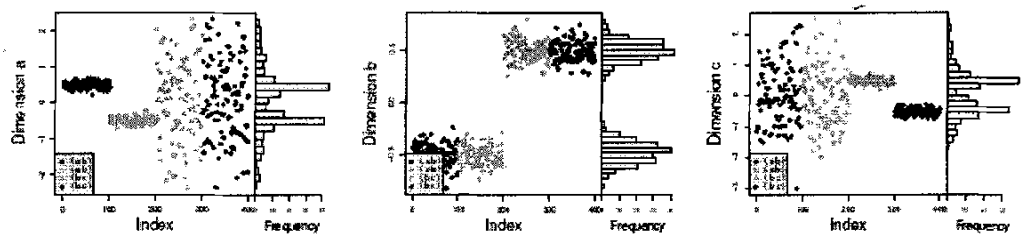


图 3.1 样例数据集的分布

从图 3.1 中我们可以看出，在不相关的属性特征上，聚类数据将是展开的，类似 k-means 的方法运用在这类数据集上将得到非常差的结果。在高维的情况下，这个问题将变得更加严重，甚至将不可能发现聚类，因此我们必须考虑在较低维上进行聚类。

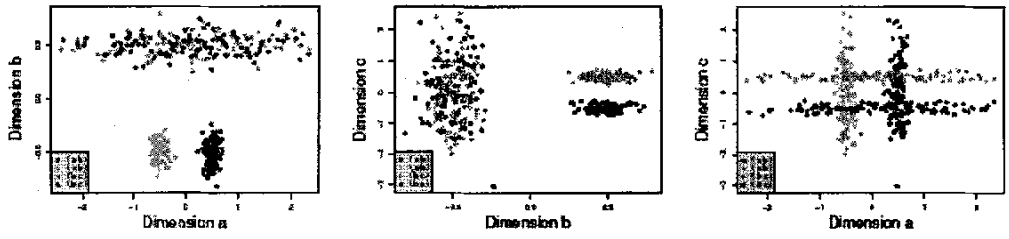
正如上一节讨论的，特征转换方法由于使用了相对距离的概念使得不相关属性维的影响遗留下来，因此同样不适用于这类情况。下面，我们深入讨论特征选择方法是否适用的问题。



(a) Dimension a (b) Dimension b (c) Dimension c

图 3.2 样例数据集投影到一维上的直方图

如图 3.2 所示所有数据被分别投影到各个属性维上，从图 3.2 中可知，任何一种投影都不可能把所有 4 个聚类区分开来。如果只是简单的去掉其中的某一维，如图 3.3 所示我们会看到不可能得到有效的将 4 个聚类完全分开的数据投影。



(a) Dims *a* & *b*　(b) Dims *b* & *c*　(c) Dims *a* & *c*

图 3.3 样例数据集投影到任意两维上

但我们可以发现，这样的投影并不是毫无价值的。虽然图 3.3(c)不能将任意一类与其他类区分开来，但其中的两个类可以通过图 3.3(a)很容易的区分开来，另外两个类也可以通过图 3.3(b)图区分开来。这也就是子空间聚类的关键之处，进而可以在一个恰当的子空间中发现某个聚类。

3.3.2 子空间聚类算法

现有的子空间聚类算法大体上可以分为两类：自底向上（bottom-up）和自顶而下(top-down)。自底向上的算法利用了聚类对于维数的单调性的性质来减小搜索空间。具体算法有 CLIQUE，ENCLUS，MAFIA，CBF 等。自顶而下算法有 PROCLUS，FINDIT，δ -Clusters，COSA 等。本文研究的算法主要是基于 CLIQUE 算法的改进，下面我们具体介绍 CLIQUE 算法。

3.4 CLIQUE 算法

CLIQUE 算法[3]是人们提出的第一个子空间聚类算法，它综合运用基于密度和基于网格方法优点来构造聚类方法。CLIQUE 算法不仅以网格来划分数据集，而且进一步利用了数据在网格中分布的不均匀性， 通过预设阈值把网格单元划分为“稀疏的”和“稠密的”两类，并集中精力考虑稠密网格单元数据的聚类问题。

设 $A=\{A_1,A_2,\cdots,A_d\}$ 是 d 个有界定义域，那么 $S=A_1\times A_2\times\cdots\times A_d$ 就是一个 d 维空间，我们将 $A_1,A_2,\cdots,A_n$ 看成是 S 的维（属性、字段）；算法的输入是一个 d 维空间中的点集,设

为 V={$v_1,v_2,\cdots,v_n$}，其中 v_i={$v_{i1},v_{i2},\cdots,v_{id}$}。$v_i$ 的第 j 个分量 $v_{ij}\in A_j$；通过一个输入参数 ξ，可以将空间 S 的每一维分成相同的 ξ 个区间，从而将整个空间分成了有限个不相交的类矩形单元(units)，每一个这样的矩形单元可以描述为 {$u_1,u_2,\cdots,u_d$}，其中 $u_i=[l_i,h_i]$为 A_i 上的一个分割区间。

设若一个 v={v1,v2,…,vd} 落入一个区间 u={$u_1,u_2,\cdots,u_d$}，当且仅当对于每一个 u_i 都有 $l_i<=v_i<h_i$ 成立。我们定义一个单元格 u 的选择率 selectivity(u)为

$$selectivity=\frac{\text{单元格中点的个数}}{\text{数据空间中总的点的个数}} \tag{3.1}$$

对于用户的输入参数 τ，我们称数据单元 u 是密集(dense)的，当且仅当 selectivity(u)>τ;对于 S 的任何子空间，例如子空间 Sub=$A_{t1}\times A_{t2}\times\cdots\times A_{tk}$，（k<d，并且当 i<j 时有 ti<tj 成立），可以在该子空间中定义单元格，选择率等相同概念。

一个聚类可以定义为，在 k 维空间中由一些连通的密集单元格组成的连通分支，两个 k 维中的单元格 u1,u2 称为连通的（connected）当且仅当：（1）这两个单元格有一个公共的面；或者（2）u1,u2 都跟另一个单元格 u3 连通；两个单元格 u1={$R_{t1},R_{t2},\cdots,R_{tk}$},u2={$R'_{t1},R'_{t2},\cdots,R'_{tk}$} 有一个公共的面是指，存在 k-1 个维度（不妨设这 k-1 维就是 t1,t2,…,tk-1），有 $R_{tj}=R'_{tk}$ 成立（j=1,2,…,k），并且对于第 tk 维有 $h_{tk}=l'_{tk}$，或者 $h'_{tk}=l_{tk}$ 成立。

CLIQUE 采用 DNF 表达式来描述得到的聚类，为此，这里先说明区域（region）的概念，所谓区域是指一个每一边都与坐标轴平行的类矩形。也就是说这类区域是由单元格组成的且具有规则的形状，这样一个区域就可以用区间的交的形式表示出来。例如下面的表达式就表述了一个两维空间中这样的区域：

$$(30\leqslant AGE<50)\wedge(4\leqslant SALARY<8) \tag{3.2}$$

称区域 R 包含于一个聚类 C，当且仅当 $R\cap C=R$；进一步称这样的 R 是最大的（maximal）当且仅当没有一个 R 的超集 R'也包含于 C;一个聚类 C 的最小描述是上述“最大区域（maximal region）”的一个集合 r，r 中的最大区域刚好覆盖 C，并且为了满足覆盖的条件，集合 r 中的最大区域的个数不能再减少了。可将上述定义用两个例子解释如下：

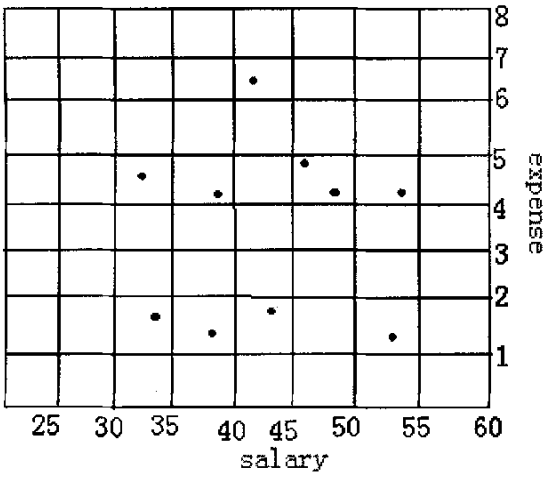


图 3.4 二维空间中的点的分布

在图 3.4 中，显示了一个二维空间中的点的分布情况。显然有 $\xi=8$，即每一维被分成了 8 个区间。如果定义 $\tau=0.4$，那么没有哪一个二维空间中的单元格是密集（dense）的，因为单元格中具有最大的 selectivity 的是单元格（45≤salary<50）∧(4≤expense<5)，它的 selectivity=2/10=0.2，同样在子空间 salary 中也没有密集的单元格，而在一维子空间 expense 中单元格[1，2]，以及[4，5]都是密集的；

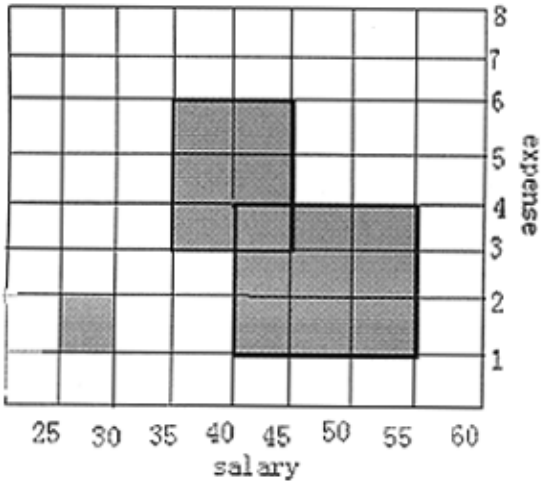


图 3.5 聚类区域的描述

在图 3.5 中，假定上述阴影部分代表密集的单元格（dense units），那么这个两维子空间中有两个聚类，分别可以描述为：(25≤salary<30)∧(1≤expense<2)，和(40≤salary<55)∧(1≤expense<4)∨(35≤salary<45)∧(3≤expense<6)。注意这里的描述没有用到区域(40≤salary<45)∧(1≤expense<6)，虽然它也是一个 maximal region。

CLIQUE 算法的主要不足在于，（1）它根据用户输入的参数等宽分割每一维，这样导致可能有某一类被人为地分割成多个区域，而在覆盖相连的密集单元时又将其相连。这使得划分单元的数目增加，在高维情况下，相邻单元的数量以指数级增长，在覆盖相连阶段将花费了大量的时间；（2）基于最小描述长度的剪枝算法，虽然提高了算法的运行效率，但同时也丢失了一些很重要的类；（3）没有提供一种基于有限内存的大规模数据集的聚类策略。

3.5 本章小结

本章讨论了高维数据聚类所遇到了问题以及各种降维处理方法，同时分析了子空间聚类的原理和优势，最后详细介绍了已有的子空间聚类算法—CLIQUE 算法，指出了 CLIQUE 算法的若干不足。下一章，将根据以上不足提出 CLIQUE 的改进算法。

第四章 基于最优分割的子空间聚类算法

本章在 CLIQUE 算法的基础上，提出一种改进的子空间聚类算法—OpCulster，以解决 3.4 节提出的问题。

4.1 OpCluster（Optimal intervals-based Cluster）算法

4.1.1 OpCluster 的算法思想

基于子空间聚类算法 CLIQUE，我们对它进行改进，设计并实现了 OpCluster 算法，其特点主要反映在以下三个方面：（1）根据数据在每一个维上的分布特征，采用文献[10]中的最优分割面的技术将每一维进行区间划分，显著减少了每维上分割的区间数和候选聚类区域集的数目，同时算法将不需要进行覆盖相连的密集单元的步骤；（2）由于候选聚类区域集减少，算法中不再采用基于最小描述长度的剪枝算法，提高了精度；（3）基于内存的聚类策略，采用划分数据集的方法，使每个划分的数据集大小适合内存，通过发现局部聚类的方法发现全局聚类，可进一步提高算法的效率。

OpCluster 算法的主要思路是在每一个维上，将样本向坐标轴上投影，利用最优分割区间算法将样本进行区间分割。由此根据样本的分布特性进行的区间分割，将比等宽分割得到更精确的聚类区间和更少的区间数目。对获得的最优分割区间，再采用基于数据集划分的聚类发现算法，得到基于子空间的聚类。

4.1.2 最优分割区间

1. 确定区间分割的原则

(1) 必须在样本密度较低的地方进行分割；

(2) 每一个分割必须尽可能的将不同聚类分开。

以上（1）保证了不会将一个聚类分割到多个区间，（2）保证了每一个分割对聚类都是有贡献的[11]。

2. 最优分割区间主要思路：

最优分割区间的大小依赖样本数据在这一维上的分布特性。根据文献[19]，可以得到能够较精确的反映分布特性的最少间隔数目，用以建立直方图。对于每一个直方图，我们采用

爬山的方法，从第 1 个间隔开始，相邻间隔内的样本数进行比较，如果第 j 个间隔内的样本数小于等于第 j+1 个间隔内的样本数，则是在爬山，否则是在下山。通过以上方法，我们确定下每一个山谷位置。针对每一个山谷并且它是夹在一对山峰中间的，将进行下面的假设检验[10]。

$$\chi^2 = \frac{2(observed - expected)^2}{expected} \geq \chi^2_{\alpha,1}, \qquad (4.1)$$

其中值 observed 等于直方图中山谷的取值，值 expected 等于直方图中山谷取值和较低那个山峰取值的平均值。在实际运用中，通常取 α 为 95%，置信水平（$\chi^2_{0.05,1} = 3.843$）。

对满足公式（4. 1）的每一个山谷进行的有效分割，我们引入最小密度阈值 ρ 。对于每一个区间，如其密度大于密度阈值，则该区间被认为是某一聚类在该维上的投影区间，即一维的聚类区域。

3．最优分割区间算法

算法：最优分割区间算法

输入：DB　:大规模、高维数据库，数据元组总数为|DB|，维数为 d

　　ρ　：最小密度阈值

输出：set_i（1≤i≤d）:在第 i 维上聚类投影区域集

步骤：

for each dimension A_i 1≤i≤d do begin

set_i= Φ;

计算最小间隔数目，建立直方图；

采用爬山的方法，确定每个山谷的位置；

针对每一个山谷并且它夹在一对山峰中间，利用公式（4. 1）找到有效分割；

计算每个区间密度，若大于 ρ，则得到一个有效区间 a_{ij}，$set_j = set_i \cup a_{ij}$；

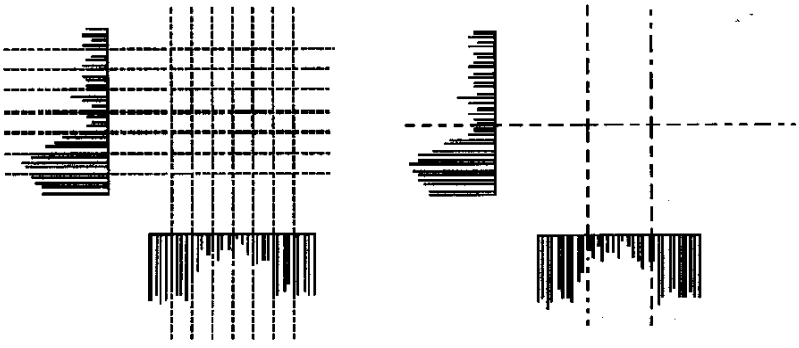
end.

4．最优分割区间对聚类的影响

如图 4. 1 所示，CLIQUE 所采用的平均划分区间方法将会在聚类发现的每一步过程中，比最优分割区间的方法产生更多的候选聚类项集。CLIQUE 采用贪心算法来生成相连密度单元的最小覆盖，产生 DNF 范式。不过在减小复杂度同时降低了聚类的精度。

OpCluster 采用的最优区间分割和假设检验的方法完全根据数据的分布特性，减少了分

割区间数目。OpCluster 不需要生成最小覆盖的过程，其结果为 DNF 范式。



(a)等宽区间分割　　(b)最优区间分割

图 4.1　最优分割区间对聚类的影响

4.1.3 对大规模数据的高效处理

根据最优分割区间算法，可以得到在每一维上的一维的聚类区间。CLIQUE 算法在每一步生成 k 维子空间聚类的过程中都要扫描一次数据集。对大规模数据集来说，这显然影响了算法的效率。OpCluster 采用划分数据集，减少 I/O 访问的次数，通过发现局部聚类区域来发现全局聚类区域的方法以提高算法的执行效率。

1．相关概念和结论

定义 4.1　设 M 为内存可存数据项的数目，|DB|为数据集中所有数据元组数目，按照可用内存的大小将 DB 划分为多个不相交的数据集 $\{p_i\}$，则划分数目 $N=\left\lceil\frac{|DB|}{M}\right\rceil$，其中 $|p_1 \cup p_2|...\cup p_{N-1}|=M, |p_N|\le M$。

定义 4.2　若 k(2≤k≤d, d 为向量空间的维)维向量子空间区域 X 为某一样本类 C 所有样本的投影区域，则称 X 为全局 k 维子空间聚类区域，若 X 为 C 在划分 $p_i(1\le i\le N)$ 上所有样本的投影区域，则称 X 为局部 k 维子空间聚类区域。

引理 4.1　若 X 为局部 k 维子空间聚类区域，则 X 的 k-1 维投影区域均为局部 k-1 子空间聚类区域。

证明：用反证法证明，设 X 的任意 k-1 维投影区域 S_{k-1} 不是局部子空间聚类区域，则在 S_{k-1} 中的样本数 $n_{k-1}<|p_i|\times\rho$，又因为 $S_{k-1}\subset X$，所以 X 中的样本数 $n_k\le n_{k-1}<|p_i|\times\rho$，所以 X 不是 k 维子空间聚类区域。与题设矛盾，引理得证。

推论 4.1 若 X 不是局部 k 维子空间聚类区域，则 X 的 k+1 维扩展区域一定不是局部 k+1 维子空间聚类区域。

类似地，若 X 不是全局 k 维子空间聚类区域，则 X 的 k+1 维扩展区域一定不是全局 k+1 维子空间聚类区域。

定理 4.2 若 X 为全局 k 维子空间聚类区域，则至少存在一个划分 $p_i(1 \le i \le N)$，使得 X 和 X 的所有 k-1 维投影区域在该划分上为局部子空间聚类区域。

证明：用反证法，设不存在一个划分 $p_i(1 \le i \le N)$，使得 X 在该划分上为局部子空间聚类区域，即某一样本类 C 在划分 p_i 上在 X 区域内的样本数 $n_i < |p_i| \times \rho$，$1 \leqslant i \leqslant N$。则 $\sum_{i=1}^{N} n_i < (\sum_{i=1}^{N} |p_i|) \times \rho = |DB| \times \rho$，即 X 不是全局子空间聚类区域。与题设矛盾，定理得证。

定义 4.3 对任一数据集划分 $p \subseteq DB$，其中 $p_i \cap p_j = \phi, i \neq j$，DB 为数据库，有以下定义：

L_k^p：在划分 p 上的局部 k 维子空间聚类区域集合；

L^p：在划分 p 上的局部子空间聚类区域集合；

C^g：候选全局子空间聚类区域集合；

D^g：全局子空间聚类区域集合。

2. 基于样本划分的聚类发现算法

算法：基于样本划分的聚类发现算法

输入：DB ：大规模、高维数据库，数据总数目为|DB|，维数为 d

M ：可用内存

ρ ：最小密度阈值

set_i（$1 \leqslant i \leqslant d$）：由最优分割区间算法得到的各维聚类投影区间集

输出：D^g：全局子空间聚类区域集合

步骤：

p=patition_dabase(DB) $p_i \cap p_j = \phi, i \neq j$；

$N=\left\lceil \frac{|DB|}{M} \right\rceil$, $|p_i|=M, i=1..N-1$, $p_N \le M$;

for j= 1 to N do　begin

将数据块 p_j 读入内存;

$L_1^{p_j}=\bigcup_{i=1..d} set_i$;

for(k=2; $L_k^{p_j} \neq \phi$;k++) do begin

for all $l_1 \in L_{k-1}^{p_j}$ do begin

for all $l_2 \in L_{k-1}^{p}$ do begin

if(l_1 与 l_2 中任意 k-2 维区域相同而另一维区域不同)

then $c=l_1[1]\cdot l_1[2]\cdots l_1[k-1]\cdot l_2[i]$ //其中 $l_2[i]$ 是与 l_1 中各维不同的那维

//上的区域;

if (c 的所有 k-1 维子投影区域包含于 $L_{k-1}^{p_j}$)

and (数据块 p_j 在 c 上的投影密度> ρ),

then $L_k^{p_j}=L_k^{p_j}\cup\{c\}$;

end;

end;

$L^{p_j}=L^{p_j}\cup L_k^{p_j}$;

end;

$C^g=\bigcup_{j=1..d} L^{p_j}$;

扫描一次数据库 DB，对所有 $c \in C^g$，计算 c 的密度，若大于 ρ，则 $D^g=D^g\cup\{c\}$.

输出全局聚类区域 D^g。

4.2 算法性能分析

由于 OpCluster 是在 CLIQUE 算法基础上加以改进，因此 OpCluster 具有 CLIQUE 相关优点：较强的噪声处理能力、对输入样本的顺序不敏感、可以发现任意形状的聚类、聚类结果易于理解等。

相比 CLIQUE 算法，OpCluster 还具有以下优点：（1） 较低的时间复杂度。若数据中存在最高维的聚类其维数为 k，数据量维 m，则 OpCluster 最坏情况下的时间复杂度为 $O(c^k+3m)$，而 CLIQUE 的复杂度为 $O(c^k+km)$。因为减少了坐标轴上的聚类区域，一方面，聚类子空间的数目显著减少，同时省去了覆盖相邻密度单元的步骤。（2）较好的可扩展性。OpCluster 对大规模数据有很好的可扩展性，当数据增多时，数据分布特性更加明显，OpCluster 可以获得更好的聚类效果。数据越多，OpCluster 相比 CLIQUE 效率越高。（3）较高聚类结果的精确度。基于最优化分割区间的算法能够更加精确的刻画聚类的边界，不采用剪枝算法，得到的聚类结果更加精确。

4.3 实验与结果分析

我们使用多个仿真数据集来比较 OpCluster 与 CLIQUE 算法的性能差别。实验平台配置如下：Intel 1.8G/256MB，Windows2000(Server 版)，所用代码均用 Visual C++(6.0)实现。

4.3.1 仿真数据的产生

实验中所使用的仿真数据产生器是根据 DataGen[29]改进而成。其产生的仿真数据类似于 CLIQUE 算法中所用的数据集。用户可以通过输入参数来控制产生数据集的结构与大小。参数包括数据集的大小、维数和各维上的取值范围（实验中为[0，100]）等。

4.3.2 实验结果

1. 在图 4.2 中，数据集的大小从 100000 增大到 500000。数据空间维度为 50，在不同的 5 维子空间中存在 5 个聚类。分析图 4.2 可以看到，在相同维度不同数据规模的情况下 OpCluster 的执行效率明显比 CLIQUE 高。

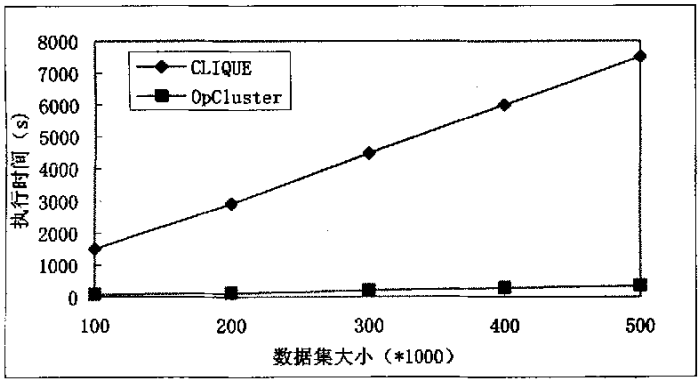


图 4.2 不同数据集大小的性能比较

2．图 4.3 中数聚集的数据空间维数由 10 到 100，数据集大小为 10000，在不同的 5 维子空间中存在 5 个聚类。分析图 4.3 可以看到，在相同数据集规模的情况下 OpCluster 对数据空间维度有较好的伸缩性。

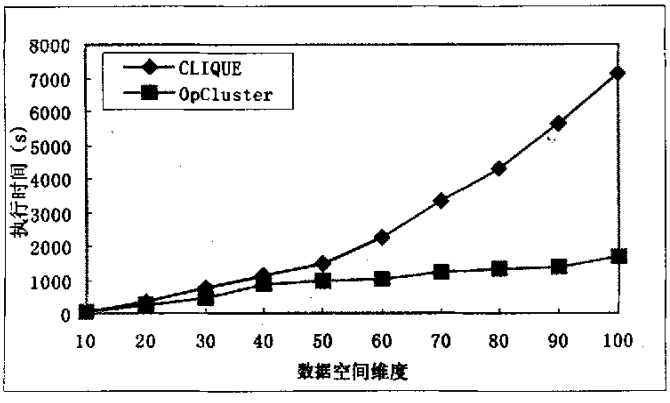


图 4.3 不同数据空间维数的性能比较

3．在图 4.4 中，数据集中存在的聚类最高维从 3 增大到 10。数据集数据空间维数为 50，大小为 20000。从图 4.4 中可知，OpCluster 具有与 CLIQUE 相似的伸缩性，性能略有改善。

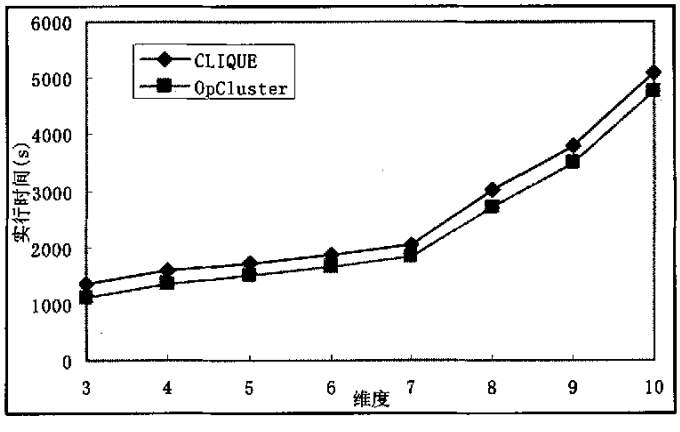


图 4.4 不同最高聚类维数的性能比较

4.4 本章小结

本章提出了一种基于最优分割区间的子空间聚类算法 OpCluster。它能够有效的改善聚类算法对数据集大小和数据空间维数的伸缩性，同时聚类结果的精度也有相当大的提高。与 CLIQUE 算法的实验比较可以证明，OpCluster 算法是有效的。但从实验分析中可看出，对于存在高维聚类的情况，OpCluster 算法只能说相比 CLIQUE 性能有所改善。下一章，将提出基于递减阈值函数的有趣子空间发现算法，对于存在高维聚类的情况，效率有很大提高。

第五章 有趣子空间区域发现算法

在上一章中，我们提出了基于 CLIQUE 算法的改进算法—OpCluster。但从实验分析中，我们发现当数据集中存在高维聚类时，OpCluster 算法效率仍然不是很理想。在本章中，我们受文献[23]启发，将提出一种新的聚类发现算法，通过减少低维候选项数目来提高算法效率。

5.1 有趣的定义

定义 5.1 若一个子空间区域是有趣的，则在该子空间区域内样本点的数量在统计意义下明显高于期望值，其中期望值是样本点在所有属性维上独立同分布的假设下得到。

显然，从聚类的观点来看，如果所有数据点平均散布到所有属性维上有趣度将是最低的。如果在一个子空间区域内，样本分布明显高于平均分布，那么该子空间区域将有可能是有趣的，即有可能有聚类存在。

若一 p 维子空间区域S_p，在该子空间区域内有n_p个样本点，则在前述 CLIQUE 算法中，如果$n_p/n \geq s$，其中 s 为用户定义的最小支持度阈值，该子空间区域是密集的。在 MAFIA 算法中，如果$n_p / n \geq (\alpha a) / D_i$，其中 α 为用户定义的聚类因子，则该子空间区域是密集的。总之，在基于密度的子空间聚类算法中，都需要一个阈值函数$f: Z^+ \rightarrow \Re$，Z^+为正整数集合，表示候选子空间聚类区域的维数，$\Re$为该子空间区域如果是密集所必须要超过的最小阈值，例如在 CLIQUE 算法中$f(p)=s, \forall p \in [1..d]$，在 MAFIA 算法中$f(p)=(\alpha a)/D_i, \forall p \in [1..d]$。对于阈值函数$f$可以是常数（如 CLIQUE），也可以是单调递增或单调递减函数。

定理 5.1 若任一(p+1)维子空间区域$S_{p+1} \subset S$是有趣的，则任一S_{p+1}的投影区域S_p都是有趣的，当$f(p+1) \geq f(p), p \in [1..d-1]$。

证明：若S_{p+1}有趣，则$\frac{n_{p+1}}{n} \geq f(p+1)$，因为$S_p$是$S_{p+1}$的投影区域，所以$n_p \geq n_{p+1}$。

所以$\frac{n_p}{n} \geq \frac{n_{p+1}}{n} \geq f(p+1) \geq f(p)$。根据定义 5.1，$S_p$是有趣的。

定理 5.2 若阈值函数f是单调非递减函数，则任一 p 维子空间区域$S_p, p \in [1...d]$，在域值$f_2(p) = f(r)$下是有趣的，其中 r 是在f下的最大维的子空间有趣域区S_r的维数，且$S_r \subseteq S_p$。

证明：因S_r是有趣的，则$\frac{n_r}{n} \geq f(r)$，又因为$S_r \subseteq S_p, 1 \leq p \leq r-1$，则$n_p \geq n_r$。

因此$\frac{n_p}{n} \geq \frac{n_r}{n} \geq f(r) = f_2(p)$。所以$S_p$是有趣的。

根据定理 5.1 和定理 5.2 可知，在高维数据集中，若存在一个最高维的子空间有趣区域S_r，则为了得到该有趣区域，我们必须把$f(1)$定的尽可能的小（因为必须$f(1) \leq f(r)$），在聚类的低维阶段，就会产生大量的候选项集，这将显著降低算法的时空效率。因此为了提高算法的效率，主要是减少在低维聚类阶段产生的候选项集的数目。同时我们可以采用递减的阈值函数f，当在低维时取较高的阈值，能够裁减掉大量候选项集。这也是符合事实规律的，显然随着维数的增加，在任一子空间区域的数据样本数也越来越少。

定义 5.2 若子空间区域S_p中含有n_p个样本点的概率$\Pr(X_p \geq n_p) \leq \gamma$，则$S_p$是有趣的，其中$\gamma$是用户给定的概率阈值。

5.2 Chernoff-Hoeffding 界

本章中，我们将使用 Chernoff-Hoeffding 界[17]来描述X_p分布的界，并用它来度量S_p的有趣性。

Chernoff-Hoeffding 界：设$X_i, i = 1...n$，是独立同分布随机变量，其中$0 \leq X_i \leq 1, Var[X_i] < \infty$，则对于$X = \sum_{i=1}^{n} X_i, t > 0$，

$$Pr[X \geq E[X] + nt] \leq e^{-2nt^2} \tag{5.1}$$

设S_p为一 p 维子空间区域，X_p表示数据集DB中的样本点中投影到S_p的样本点，采

用公式（5.1）中的界，得若S_p是有趣的，则

$$Pr[X_p \geq n_p] \leq e^{-2nt_p^2} \leq \gamma \tag{5.2}$$

比较公式（5.1）和（5.2）得$E[X_p]+nt_p=n_p$，即$t_p=\frac{n_p}{n}-\frac{E[X_p]}{n}$。将$t_p$代入（5.2）得$e^{-2n(\frac{n_p}{n}-\frac{E[X_p]}{n})^2} \leq \gamma$，可得到

$$\frac{n_p}{n} \geq \frac{E[X_p]}{n}+\sqrt{\frac{1}{2n}ln(\frac{1}{\gamma})} \tag{5.3}$$

由此，我们定义有趣性的度量阈值函数为$f(p)=\frac{E[X_p]}{n}+\sqrt{\frac{1}{2n}ln(\frac{1}{\gamma})}$，其中$f(p)$是随维数$p$非线性单调递减的函数。我们也可以把$f(p)$定义为密度阈值用于 CLIQUE 算法和 MAFIA 算法，如在 CLIQUE 算法中，可令$s=\sqrt{\frac{1}{2n}ln(\frac{1}{\gamma})}$。

如果假设所有样本点在各维上是独立同分布的，类似 CLIQUE 算法，每一维都被平均分成ξ个区间，于是对一个样本点在某一维上落入一个特定区间的概率为 1/ξ。因此，一个样本点落入某一 p 维子空间区域的概率为$(1/\xi)^p$。在任一 p 维子空间区域S_p发现n_p个样本点的概率服从二项式分布，即$E(X_p)=n(\frac{1}{\xi})^p$。所以，当$\frac{n_p}{n} \geq \frac{1}{\xi^p}+\sqrt{\frac{1}{2n}ln(\frac{1}{\gamma})}$时，$S_p$是有趣的。

若采用类似第四章中最优分割区间算法对每一维进行区间划分，那我们将得到基于最优分割区间的有趣度度量函数。

定义 5.3 A= $\{A_1,A_2,\cdots,A_d\}$ 是 d 个有界定义域，那么 $S=A_1\times A_2\times\cdots\times A_d$就是一个 d 维空间，$|A_i|$为属性 A_i 的域，$a_{i1},a_{i2},\ldots a_{ik}$把 A_i 分为 k（k 为任一自然数）个区间，$|a_{ij}|$为区间 a_{ij} 的域，则$|a_{i1}|+|a_{i2}|+\ldots+|a_{ik}|=|A_i|$。$S_p=\prod_{i=1,i\in D_p}^{p}a_{ij}$ 表示一个 p 维的子空间区域，其中D_p表示S_p中所含的各维，a_{ij}表示S_p中第 i 维上的第 j 个区间。

假设所用样本点在各维上是独立同分布的，则对于一个样本点在某一维上落入一个特定区间的概率为$\frac{|a_{ij}|}{|Ai|}, i\in[1..d], j\in[1..k]$。于是一个样本点落入某个子空间区域$S_p$的概率

为 $\prod_{i\in D_p}\frac{|a_{ij}|}{|A_i|}$。在任一 S_p 中发现 n_p 个样本点的概率同样服从二项式分布，所以 $E(X_p)=n\prod_{i\in D_p}\frac{|a_{ij}|}{|A_i|}$。

定义 5.4 对于定义 5.3 中定义的子空间区域，当

$$\frac{n_p}{n}\geq\prod_{i\in D_p}\frac{|a_{ij}|}{|A_i|}+\sqrt{\frac{1}{2n}ln(\frac{1}{\gamma})} \tag{5.4}$$

时，S_p 是有趣的。

当然，如果我们采用单调递减的阈值函数，类似 CLIQUE 算法中的性质就不成立了。如 $\frac{n_p}{n}=\frac{n_{p+1}}{n}=f(p+1)$，如果 $f(p+1)<f(p)$，那么 $\frac{n_p}{n}<f(p)$。虽然子空间区域 S_p 不是有趣的，S_{p+1} 却是有趣的。但如果要在大规模高维数据集中发现高维聚类，有时候是非常耗时的。采用单调递减的阈值函数虽然不能保证发现所有有趣的子空间区域，但是可以使挖掘的时间效率更趋合理。由于公式（5.4）得到的阈值函数是由 Chernoff-Hoeffding 界获得的，因此可以认为误差是能够接受的，后面的实验也证明了这一点。

5.3 FIS 算法

算法：FIS 算法（An Algorithm for Finding Interesting Subspace Using Dimension-Decreasing Support Constraint）

输入：DB ：大规模、高维数据集，数据元组总数为|DB|，维数为 d

γ ：概率阈值

输出：C ：有趣子空间区域

步骤：

for each dimension A_i $1\leqslant i\leqslant d$ do begin

计算最小间隔数目为 T，将 DB 中的数据投影到 A_i，建立直方图；

确定直方图的范围为$[L_i,H_i]$；

j=1；

for p=2 to (T-1) do begin

q=1;

if (value[p-1]>value[p]) and (value[p]≤value[p+1])

then a_{ij}=[value[q],value[p]];//函数 value 求直方图的值

q=p;j=j+1;

end;

a_{ij}=[value[q],value[t]];// a_{ij} 表示第 i 维上的第 j 个区域取值

k=j;

end;

$C_1 = \phi$;// C_k 表示 k 维有趣子空间区域的集合

for i=1 to d do begin

for j=1 to k do begin

if $\frac{n_1(a_{ij})}{n} \geq \frac{|a_{ij}|}{|A_i|} + \sqrt{\frac{1}{2n} ln(\frac{1}{\gamma})}$ then $C_1 = C_1 \cup a_{ij}$;

end;

end;

$C = \phi$;

for(k=2; $C_k \neq \phi$;k++) do begin

for all $l_1 \in C_{k-1}$ do begin

for all $l_2 \in C_{k-1}$ do begin

if(l_1 与 l_2 中任意 k-2 维区域相同而另一维区域不同)

then $c = l_1[1] \cdot l_1[2] \cdots l_1[k-1] \cdot l_2[i]$; // $l_2[i]$ 是与 l_1 中各维不同的那维区域

if (c 的所有 k-1 维子投影区域包含于 C_{k-1})

and ($\frac{n_1(c)}{n} \geq \prod_{i \in D_p} \frac{|a_{ij}|}{|A_i|} + \sqrt{\frac{1}{2n} ln(\frac{1}{\gamma})}$)

then $C_k = C_k \bigcup \{c\}$;

end;

end;

$C = C \cup C_k$;

end;

end.

5.4 实验与结果分析

实验所使用的软硬件环境与第四章完全一样，我们将使用多个仿真数据集来检测 FIS 算法的性能，包括两个评价指标：（1）执行时间：在数据集中发现所有有趣子空间区域的时间；（2）精度：在本实验中，用熵作为反映精度的指标。对于一个聚类过程C，熵定义为 $\mathrm{E}(C) = -\sum_{C_j}(\frac{n_j}{n}\sum_i p_{ij} log(p_{ij}))$，其中 $p_{ij} = \frac{n_{ij}}{n}$，$C_j$ 为 C 中得到的第 j 个聚类区域。n_j 为 C_j 中数据点的数目，n_{ij} 为属于子空间区域 i 且在 C_j 中的数据点的数目。

5.4.1 仿真数据的产生

实验中所使用的仿真数据产生器与第四章相同。用户可以通过输入参数来控制产生数据集的结构与大小。参数包括数据集的大小、维数和各维上的取值范围（实验中为[0，1000]）等。

5.4.2 实验结果

1．数据集大小对 FIS 精度的影响：图 5.1 中，数据集的大小从 100000 增大到 500000。数据空间维度为 50，在不同的 5 维子空间中存在 5 个聚类。从图 5.1 中可看出，FIS 算法的精度是相当高的。

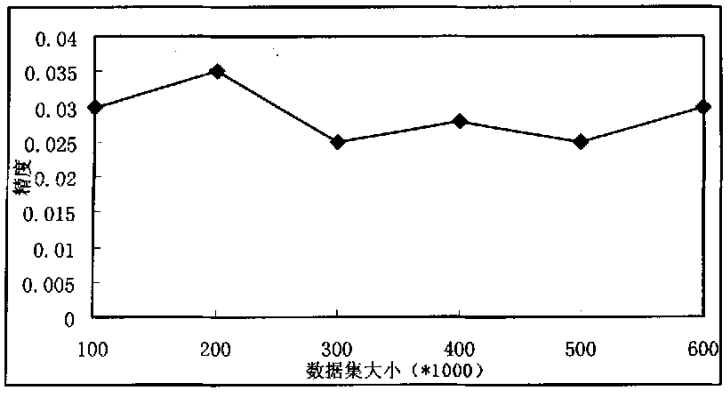


图 5.1 数据集大小对 FIS 精度的影响

2. 最高聚类维度 FIS 和 CLIQUE 执行时间的比较：图 5.2 中，数据集中存在的聚类最高维从 7 增大到 13。数据集数据空间维数为 60，大小为 20000。从图 5.2 中可知，FIS 相比 CLIQUE 性能提高是比较明显的。

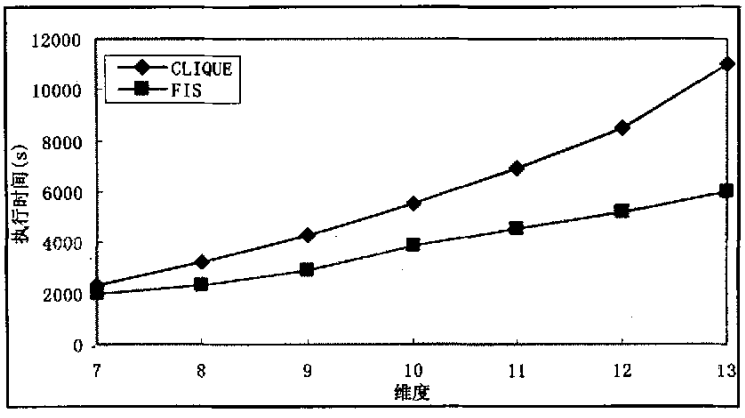


图 5.2 最高聚类维度 FIS 和 CLIQUE 执行时间的比较

5.5 本章小结

本章中提出了一种基于单调递减阈值函数的有趣子空间发现算法 FIS。它在保证精度的同时能够有效改善聚类算法对于存在高维聚类的执行效率。从实验中可以看出，FIS 算法是非常有效的。但同时也应该看到，不管是 OpCluster 算法还是 FIS 算法都不能从根本上克服“高维诅咒”，这也是值得今后进一步深入研究的子空间聚类挖掘算法的一个关键技术。

第六章 总结和展望

6.1 本文工作的总结

结合国家自然科学基金项目"基于数据挖掘技术的中观审计风险研究"课题研究的需求，我们在本论文研究工作中，主要研究了针对大规模、高维数据集的子空间聚类算法。针对大规模、高维数据集的特点，提出了 OpCluster 算法和 FIS 算法，并进行了实验验证。主要做了以下工作：

(1) 论述了数据挖掘基本概念、过程，数据挖掘应用及其当前的研究内容；

(2) 重点讨论了聚类分析方法的概念、分类及其重要价值。给出了聚类分析方法的基本过程、技术和性能评价指标。

(3) 进一步讨论了高维数据聚类所遇到的问题，传统聚类方法的在大规模、高维数据聚类时的局限性。分析了子空间聚类算法的基本思想，详细给出了已有的子空间聚类算法—CLIQUE 并提出了该算法中的若干不足之处。

(4) 针对 CLIQUE 算法的不足之处，提出了改进的 OpCluster 算法，并给出了实验验证。OpCluster 算法主要从两个方面加以改进：

① 针对 CLIQUE 算法只是根据用户输入的参数等宽分割每一维，这样导致可能有某一聚类被人为地分割成多个区域，而在覆盖相连的密集单元时又将其相连。这使得划分单元的数目增加，在覆盖相连阶段将花费了大量的时间的缺点，提出了根据数据在每一维上的分布特点最优分割区间的算法。

② 针对 CLIQUE 算法在处理大规模数据集时需要进行大量的 I/O 访问的缺点，提出了基于数据划分的聚类发现算法。

(5) 在 OpCluster 算法的实验中我们发现，当高维的数据集存在高维的子空间聚类时，CLIQUE 算法的效率很差，而 OpCluster 算法的改进也是非常有限。针对上面的问题，本文进一步给出了基于单调递减阈值函数的有趣子空间发现算法—FIS 算法。内容主要包括以下几个方面：

① 给出了子空间区域有趣的定义，利用 Chernoff-Hoeffding 界实现了有趣子空间区域的精确度量。

② 利用单调递减阈值函数，给出了有趣子空间区域的发现算法 FIS。

③ 进行了实验验证，FIS 算法在保证精度的前提下，效率提高是明显的。

6.2 需要进一步研究的问题

尽管本文提出了两个子空间聚类算法（OpCluster 算法和 FIS 算法），并证明了算法的可行性和高效性，但由于大规模、高维数据集聚类固有的复杂性以及数据挖掘在实际实现过程和今后的信息发展中可能遇到的问题，使得子空间聚类算法仍需进一步研究，主要体现在以下几个方面：

（1） 多库聚类问题的研究。在本文中，只是考虑了单个数据库的聚类问题，而多库环境给聚类算法提出了新的要求，如何实现精度与效率的统一具有很大的研究价值。

（2） 由于硬件技术的高速发展，人们获取数据的能力得到了极大的提高。现实生活中，经常可以见到这样的情况，大量需要处理的数据以很快的速度产生。处理的数据不再静态、固定存储在可多次、随机访问的介质中，而是以一种动态、流式形式出现，称其为数据流（data stream）。数据流已逐渐成为新一代计算理论与应用的研究热点之一。

（3） “高维诅咒”问题一直是高维聚类的巨大挑战，目前的聚类算法从根本说都没有完全克服这个问题。因此，对于这个问题的进一步研究将是极其重要也是极富挑战性的。

（4） 切实做好基础研究为应用服务的工作，将所取得的研究成果运用于所开展的课题研究，并在项目研究与开发工作中积极从事理论探索，促进研究工作的深入开展。

（5） 随着信息量的爆炸式增大和人们对自动挖掘的要求日益增长，自适应、无参数的聚类算法成为又一富有挑战性的课题。我们将从当前工作出发，研究和开发基于网格和密度的自适应的聚类算法。

致谢

求学的时光匆匆而过，回忆两年多来在东大的生活，我的感触很多，我要感谢生活，感谢周围的师长和身边的朋友和同学，赐予我的宝贵人生财富。

首先，要特别感谢我的导师孙志挥教授，这些年来一直给予的关心和支持。本文的完成离不开孙老师的精心指导。孙老师高尚的师德和人品、严谨的治学态度、实事求是的学术作风、勤勉敬业的工作精神、渊博的专业知识以及对学科前沿把握的敏锐性和准确性，不仅教育我、影响我，使得本论文得以顺利完成，更将对我今后的工作和生活产生深远的影响。在此，再次向我的导师敬以深深的敬意和无尽的感激！

在我的学习和生活中，得到多位学长和同学的帮助，特别感谢杨明、朱玉全、赵传申、张柏礼、孙翔、杨宜东、倪巍伟博士的指导和帮助。感谢同实验室李俊成、胡学东、李斌、尚蕾、赵艳丽、李锁花和接风华同学的帮助。同时感谢曹璟、李强等同学在生活中的帮助和支持。

还要特别感谢我的严父慈母，没有他们的严格要求和无限的关爱与照顾，也没有今天这本论文的完成。“谁言寸草心，报得三春晖”，衷心地祝父母身体健康，生活幸福！

感谢一切关心我、帮助我的人！谢谢！

参考文献

1. Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann, 2000.
2. Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS,eds. Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM Press, 1996.103-114.
3. Rakesh A, Johanners G, Prabhakar R. Automatic subspace clustering of high dimensional data for data mining application. In: Snodgrass RT, Winslett M,eds. Proc. of the 1994 ACM SIGMOD Int'l Conf. on Management of Data Minneapolis: ACM Press, 1994,94-105.
4. Hinneburg A, Keim D. An efficient approach to clustering in large multimedia databases with noise. In: Agrawal R, Stolorz PE, Piatetsky-Shapiro G, eds. Proc. of the 4th Int'l Conf. on Knowledge Discovery and Data Mining (KDD'98). New York: AAAI Press, 1998.58-65.
5. Wang W, Yang J, Muntz RR. STING: A statistical information grid approach to spatial data mining. In: jarke M, Carey MJ,Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA,eds. Proc. of the 23rd Int'l Conf. on Very Large Data Bases. Athens: Morgan Kaufmann, 1997.186-195.
6. Sheikholeslami G, Chatterjes S, Zhang AD. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In: Gupta A,Shmueli O,Widom J,eds. Proc. of the 24th Int'l Conf.on Very Large Data Bases. New York: Morgan Kaufmann, 1998. 428-439
7. Karypis G, Han Eh, Kumar V. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. Technical Report, #99-007,Department of Computer Science and Engineering, University of Minnesota, 1999.
8. S.Goil, H.Nagesh,and A.Choudhary. Mafia: Efficient and scalable subspace clustering for very large datasets. Technical Report CPDC-TR-9906-010, Northwestern University, 2145 Sheridan Road, Evanston IL 60208, June 1999.
9. C.-H.Cheng, A.W.Fu,and Y.zhang. Entropy-based subspace clustering for mining numerical data. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 84-93.ACM Press, 1999.
10. Boriana L. Milenova, Marcos M. Campos. O-Cluster: Scalable Clustering of Large High Dimensional Data Sets. In 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan 2002.
11. A.Hinneburg, D.Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In Very Large Data Base, Proceedings of the 25th International Conference on, pages 506-519, Edinburgh, 1999.
12. Beyer K., Goldstein J., Ramakrishnan R., Shaft U.: 'When is nearest neighbor meaningful?', Proc. 7nd Int. Conf. on Database Theory(ICDT), 1999.
13. Ashoka Savasere , Edward Omiecinski , Shamkant B. Navathe, An Efficient Algorithm for Mining Association Rules in Large Databases, Proceedings of the 21th International Conference on Very Large Data Bases, p.432-444, September 11-15, 1995
14. Wang JH, Shen Z, Hu YF, An applicable and efficient clustering algorithm. Journal of Software, 2004,15(5):697-705
15. Ester M, Kriegel H, Sander J, Xu XW. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han JW, Fayyad UM, eds. Proc.of the 2nd Int'l Conf. On Knowledge Discovery and Data Mining (KDD'96). Portland: AAAI Press, 1996. 226-231.

16. Ankerst M, Breuning MM, Kriegel HP, Sander J. OPTICS: Ordering points to identify the clustering structure. In: Delis A, Faloutsos C, Ghandeharizadeh s, eds. Proc. ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM Press, 1999. 49-60.
17. W.Hoeffding. Probability inequalities for sums of bounded random variables. J. American Statistical Association, 58:13-30, 1963.
18. Ordonez C, Omiecinski E. FREM: Fast and robust EM clustering for large data sets. In: Kalpakis K, Goharian N, Grossman D, eds. Proc. of the 2002 ACM CIKM Int'l Conf. on Information and Knowledge Management. McLean: ACM Press, 2002. 590-599.
19. M.P. Wand. Data-based choice of histogram bin width. The American Statistical,51:59-64,1996.
20. Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: A review. SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, 2004.
21. J.-W. Chang and D.-S. Jin. A new cell-based clustering method for large, high-dimensional data in data mining applications. In Proceedings of the 2002 ACM symposium on Applied computing, pages 503-507. ACM Press, 2002.
22. Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. In:Haas LM, Tiwary A,eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Seattle: ACM Press, 1998.73-84. http://citeseer.ist.psu.edu/guha98cure.html
23. Karlton Sequeira, Mohammed Zaki. SCHISM: A New Approach for Interesting Subspace Mining. Accepted at ICDM 2004.
24. 李存华. 基于近似密度构造的聚类分析与离群点检测算法研究[D]: [博士学位论文]. 南京：东南大学, 2004
25. Kaufman L, Rousseenw P J. Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & Sons, 1990.
26. Raymond T Ng, Han Jiawei. Efficient and Effective Clustering Methods for Spatial Data Mining(1994). In: Proceedings of 20th International Conference on Very Large Data Bases, Santiago, Chile, 1994:144-155.
27. S. Guha, R. Rastogi, and K. Shim. Rock: A Robust Clustering Algorithm for Categorical Attributes. In: Proceedings of the 15th IEEE International Conference on Data Engineering, Sydney, Australia, 1999:512-521.
28. Mannila. Methods and Problems in Data Mining. In: Proceedings of International Conference on Database Theory, Delphi, Greece, 1997.
29. Datagen http://www.datgen.com/

作者简介

基本信息

姓名：周晓云　　性别：男　　年龄：25　　籍贯：江苏省太仓市

学习经历

1998.9－2002.7　东南大学　系：计算机工程与工程　攻读学位：学士
2002.9－2004.12　东南大学　专业：　计算机应用　　攻读学位：硕士

参加的科研课题

企业商务智能决策导航器　科技部中小企业创新基金资助 2002.3－2003.9

发表的论文

- 一种基于加权的高效关联规则挖掘算法的设计与实现　第一作者
  计算机工程与应用，2004.10